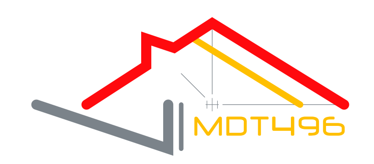


MDT496 ESTUDIOS
DEPARTAMENTO DE OBRAS LITERARIAS
COLECCIÓN: MANEJO DE PAQUETES OFFICE
LaTeX - SAS Planet - MathCAD - Maple - AutoCAD

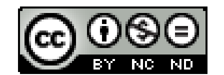
Busquedas en internet

Marco Mendieta Parihuancollo

Primera Edición

www.mdt496.wix.com/live
mdt496@gmail.com
Oruro, Bolivia

Busquedas en internet
Marco Mendieta Parihuancollo
Primera Edición (mdt496-ID: 036MP.AAAV.005)



Contenido

Lista de figuras

Capítulo 1

Introducción

Esta monografía pretende incentivar las formas de buscar, analizar y filtrar la información disponible en internet.

- Metadatos y firmas de un archivo
- Hash e integridad de un archivo
- Operadores y comandos de filtrado para las busquedas en internet

1.1 Metadatos

Son información adicional que un archivo posee para su clasificación, por ejemplo un archivo:

JPG ó una fotografia puede tener el modelo de cámara, alto y ancho de pixel, velocidad ISO, distancia focal, etc.

PDF Numero de páginas, Autor, Título, Software productor, Tipo de hoja, etc.

1.2 Hash

Un Hash es un algoritmo utilizado para crear firmas digitales y almacenamiento de passwords en una bases de datos. Técnicamente es una función de una vía, toma un mensaje (texto o archivo) de longitud variable y lo convierte en una salida de longitud fija, este proceso se conoce como digestión, como lo demuestra el código 1.1.

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import javax.xml.bind.DatatypeConverter;

public class Ejemplo {
    public static void main(String[] args) throws NoSuchAlgorithmException {
        String msg = "Hola";
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte [] digest = md.digest(msg.getBytes());
        String hash = DatatypeConverter.printHexBinary(digest);
        System.out.println(hash); //F688AE26E9CFA3BA6235477831D5122E
    }
}
```

Código 1.1. Hash MD5, programación en Java

Los hash más conocidos son: MD2, MD4, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512

1.2.1 Funcionamiento

Para ver un ejemplo práctico de como funciona un hash vease el video *¿Qué es un HASH y para que sirve?* - YouTube/EZESOLER, minuto: 04:29.457, Consultado: 2020-04-25.

1.2.2 Colisión

Es generar (con mucho computo) un hash que sea identico aunque la entrada no sea la misma.

1.2.3 Usos

Dos archivos que tengan el mismo hash implica en un 99.99% que son iguales, sabiendo esto podemos eliminar archivos duplicados de forma automática, de hecho, los servidores hacen uso de esta técnica para depurar archivos que incumplen o son denunciados por copyright.

Los recursos de una aplicación compilada, como un apk, nesesitan verificar la integridad de cada uno de sus miembros y asi ver si algun programador ajeno al equipo desarrollador insertó código, por ejemplo la evación de licencias para activación.

Las bases de datos emplean los hash para proteger las contraseñas de los usuarios, de forma que si un atacante logra tener acceso a la DB no podra saber la password. De hecho las bases de datos emplean hashes anidados, es decir, aplicar nuevamente el algoritmo hash al resultado obtenido. Como muestra vease las lineas pseudo código:

```
msg = "Hola"
resultado  = hash(msg)
resultado2 = hash(resultado)
resultado3 = hash(resultado2)
...
```

1.3 Motores de búsqueda

Existen muchos motores de búsqueda en el mundo, como ser: Google, Bing, DuckDuckGo, Baidu, Soso, Naver, Yandex, Seznam, etc.[7] Sin embargo en esta monografía solo exploraremos tres de ellos:

Google © 1998-2020 Google Inc. United States

DuckDuckGo conocido como: *el motor de búsqueda que no te rastrea*, fundado en 2008 por Gabriel Weinberg. https://duckduckgo.com/

Yandex © 2015–2020 YANDEX LLC, Rusia, https://yandex.ru

Cada uno tiene métodos propios para afinar busquedas, esto lo veremos en los capítulos que siguen.

Capítulo 2

Google

Google indexa hoy en día millones de páginas web y no entiende de zonas privadas ni restringidas. Todo aquello que esté enlazado a algún sitio y sea accesible estará en Google. Si conocemos cómo funciona Google (y en general los buscadores), podremos acceder a mucha información que ni siquiera sus propietarios saben que se encuentra ahí.[1, p.60]

2.1 Introducción

Google ofrece varios operadores de filtrado de datos. Por ejemplo con una busqueda tradicional como `tesis de postgrado en SEO y SEM` el buscador muestra mas de mil resultados, entonces es aqui donde podemos filtrar los resultados declarando que: unicamente muestre archivos PDF del portal de la Universidad Mayor de San Andres.

```
filetype:pdf site:www.soe.uagrm.edu.bo tesis AND postgrado AND SEO AND SEM
```

2.2 Operadores

2.2.1 Operador "

Busca una coincidencia exacta.

```
"postgrado en SEO y SEM"
```

Buscamos cadenas completas de texto en los resultados.

2.2.2 Operador *

Este carácter es usado como un comodín en la búsqueda exacta; con el significado de que puede aparecer cualquier palabra o palabras en lugar del comodín.

```
"Crisis * en Bolivia"
```

Otro ejemplo:

```
"postgrado en SEO *"
```

2.2.3 Operador AND

Se requiere la presencia obligatoria de estas palabras en los resultados.

```
tesis AND concreto AND hormigón
```

Por ejemplo si una búsqueda tiene varios sinónimos en vez de hacer varias búsquedas puedes hacer una sola y obtener los resultados más relevantes.

2.2.4 Operador | (OR)

Buscar cualquiera de estas palabras.

`maratón OR carrera`

Se puede usar tanto el operador OR o tambien el simbolo `|`.

2.2.5 Operador - (NOT)

Indica a google excluir todos los resultados que contengan esa palabra.

`-Roedor -"Jack Russell"`

El operador NOT se declara con el símbolo - (menos).

2.2.6 Operador @

Si lo que quieres es buscar algún término dentro de una red social, con incluir una "@" delante del término que buscamos es suficiente. Por ejemplo, si queremos buscar el perfil de Donald Trump en Twitter, se tendrá que escribir "@Twitter Trump".

@coche, @banco @libro

Sirve para buscar o encontrar etiquetas sociales asociadas con twitter ej. @nombre

2.2.7 Operador

`#marketing, #matematicas #busquedaseninternet`

Sirve para buscar términos publicados con "hastags" almohadillas en las redes sociales

2.2.8 Operdaor $

`iphone 400$, ford focus 8000$`

Sirve para buscar precios de productos en los rangos que le indiquemos

2.2.9 Operador .. (dos puntos)

`guerras 1600..1900`

Permite hacer búsquedas entre intervalos de números o fechas

2.2.10 Operador +

`+david, +gonzalo, +cocacola`

Sirve para buscar perfiles o cuentas sociales de google+ y grupos sanguíneos

2.3 Comandos

2.3.1 Atención con el uso de los comandos

Búsqueda de Google, por lo general, ignora la puntuación que no es parte de un Comando de búsqueda. No uses espacios entre el símbolo o la palabra y tu término de búsqueda. Si buscas `filetype:pdf`, la búsqueda funcionará, pero si ingresas `filetype: pdf`, no lo hará.

2.3.2 Comando define:

Sirve para buscar la definición de cualquier palabra que desconozcamos.

`define:procrastinar`

2.3.3 Comando filetype:

Sirve para localizar paginas que contenga archivos ppt, pdf, xls, doc etc... relacionados con un termino o palabra calve.

```
filetype:pdf tesis
```

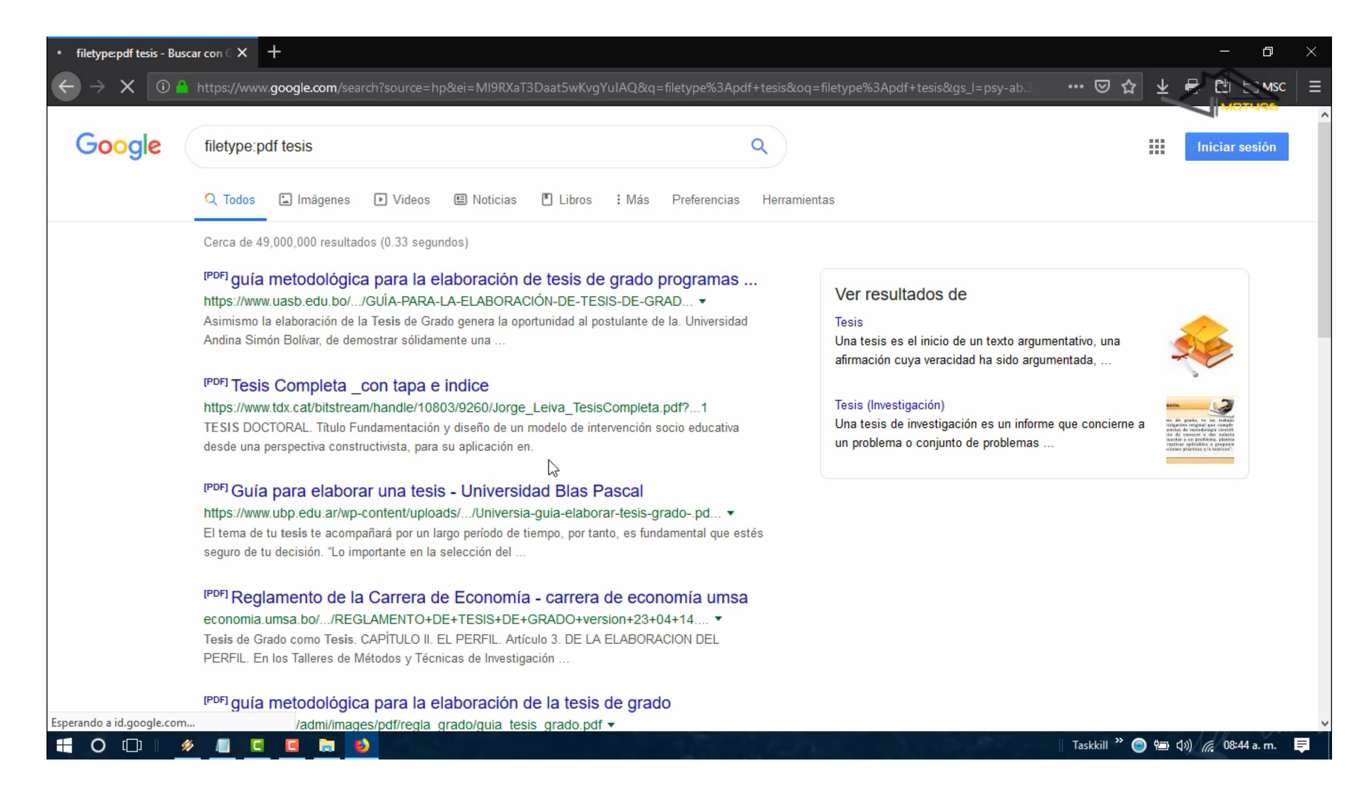


Figura 2.1. Resultado del comando filetype

2.3.4 Comando site:

Devuelve todas las páginas que el buscador ha rastreado en un dominio determinado.

```
site:www.soe.uagrm.edu.bo tesis
```

Por sí sólo no nos aporta mucha información pero podemos añadir una determinada keyword o una cadena de palabras y Google nos mostrará sólo las páginas de ese dominio que contengan los términos especificados. Es decir si queremos buscar en facebook.com que contenidos públicos existen de mi marca no tengo más que teclear `site:facebook.com IEBS` y veré todo lo que se está diciendo ahora sobre mi empresa. También es útil para buscar en páginas que no tienen buscador.

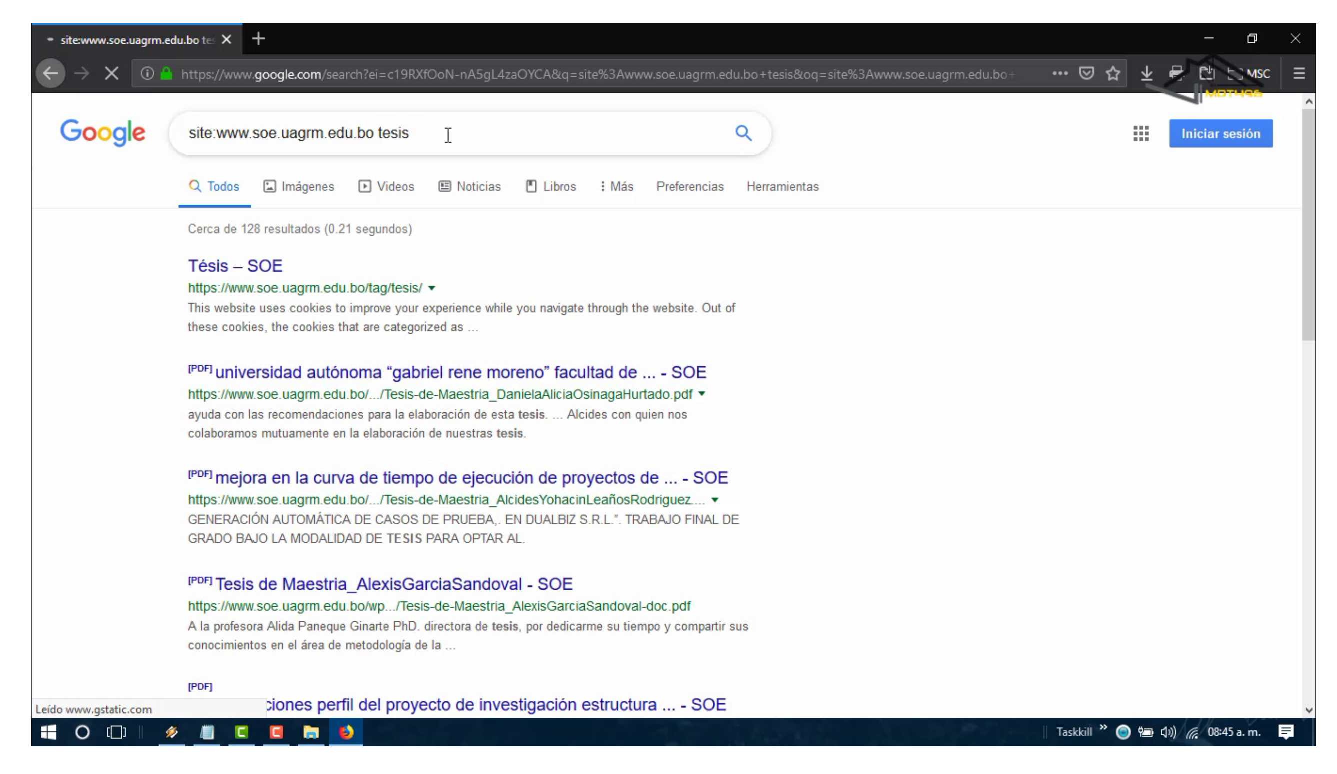


Figura 2.2. Resultado del comando site

Para extraer URLs con backlinks que mencionen la marca:

Si pones `dominio.com -site:dominio.com` en el buscador, te devuelve todos los resultados de aquellas páginas que incluyen un enlace con anchor text de marca hacia el dominio en cuestión. Funciona muy bien para verificar aquellos dominios de referencia que apuntan hacia nuestro dominio con un anchor text de marca, por si se ha capado el rastreo a los bots como el de ahrefs u otras herramientas SEO.

2.3.5 Comando domain:

Limita los resultados al dominio especificado. Se puede usar solo el identificador del país (como .es para España).

```
domain:bo
```

2.3.6 Comando related:

Sirve para localizar sitios web similares a la dirección url que se mensiona.

```
related:tesis.pucp.edu.pe
```

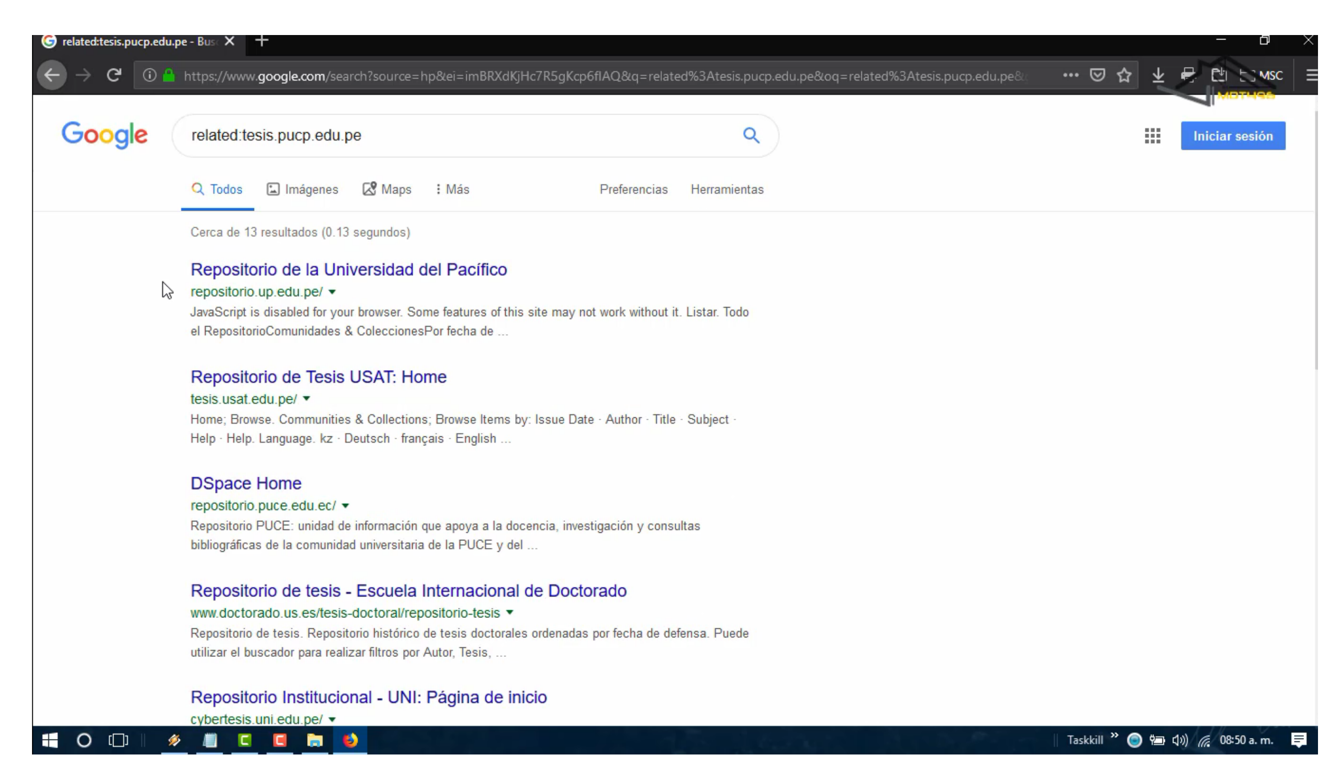


Figura 2.3. Resultado del comando related

2.3.7 Comando intitle: ó allintitle:

Permite buscar las palabras clave en el título de las diferentes páginas.

```
intitle:Repositorio de tesis
```

```
allintitle:Repositorio de tesis
```

Por ejemplo `intitle:postgrado en SEO y SEM` nos ofrecerá aquellos resultados que hay en internet que contienen en el título los términos postgrado en SEO Y SEM pero también resultados que contengan combinaciones entre ellas: SEO y SEM; postgrado en SEO, etcétera.

2.3.8 Comando inurl: o allinurl:

Realiza la búsqueda del terminio en la URL de la página.

```
inurl:blogspot
```

Sirve para detectar las páginas que contienen cualquiera de los términos empleados en la url.

2.3.9 Comando intext: ó allintext:

Busca el texto en el cuerpo de las páginas devueltas.

```
allintext:"promocion de viviendas"-madrid
```

Sirve para localizar páginas webs que contengan en su texto el termino de búsqueda deseado

2.3.10 Comando inanchor: ó allinanchor:

```
allinanchor:"seguros de coche baratos"
```

Localizar páginas web con un texto-ancla enlazado que coincida con nuestro término.

Inanchor:palabras clave nos ayudará a conocer los resultados que reciben enlaces entrantes con una, varias o todas las palabras de las que hayamos señalado en el anchor text (los textos del enlace).

Allinanchor:palabras clave es bastante similar al anterior pero se diferencia en que el texto-ancla debe contener todos los términos que indiquemos.

2.3.11 Comando info:

muestra toda la información sobre la home del dominio que indiquemos.

```
info:www.fni.uto.edu.bo
```

2.3.12 Comando link:

Sirve para localizar las paginas web que tienen enlace diriguidos a una pagina determinada.

<!– No puede verificar su funcionamiento–>

```
link:www.fni.uto.edu.bo
```

2.3.13 Comando cache:

nos lleva a la última versión que Google ha indexado sobre un determinado dominio (útil por tanto para comprobar si Google ha rastreado ya nuestros últimos cambios).

```
cache:www.fni.uto.edu.bo
```

Sirve para comprobar como era la pagina la última vez que el bot de Google indexo esa página web.

2.4 Combinación de comandos

Lo más divertido es que puedes combinar estas funciones entre si y generar consultas más complejas que te den resultados más precisos.

2.4.1 Ejemplo 1

Buscar solamente en el dominio `www.soe.uagrm.edu.bo` archivos del tipo `pdf` ya sea en su cuerpo o url el termino `tesis`.

```
site:www.soe.uagrm.edu.bo filetype:pdf intext:tesis inurl:tesis
```

2.4.2 Ejemplo 2

Por ejemplo, si te dedicas a la venta de pisos y te interesa ver blogs que sean competencia tuya, con el fin de analizarlos, puedes combinar los comandos de la siguiente manera: Pones `inurl:blog pisos`. Esto nos dará como resultado blogs en cuya dirección aparece la palabra pisos.

Ademas para encontrar blogs de Google o Blogspot agregamos `inurl:blog inurl:blogspot`

Si queremos ir más al grano con esta búsqueda y solo queremos encontrar dominios de España, podemos añadir el comando `site:.es`

Si lo que queremos es encontrar blos con las especificaciones anteriores, y además queremos filtrar con aquellos en los que podemos comentar para ver si cabe la posibilidad de dejar un enlace en el comentario, podemos filtrar de la siguiente manera añadiendo `intext:"post a comment"` o `intext:"deja un comentario"`. De esta manera nos mostrará resultados de blogs de españa que hablen de pisos y en los que podemos dejar comentarios.

```
inurl:blogspot inurl:blog "pisos" domain:bo intext:"deja un comentario"
```

2.5 Operadores obsoletos

2.5.1 Operador +

Incluir un + (más) delante de una palabra requiere la presencia obligatoria de esta palabra en las páginas devueltas.

Este operador fue remplazado por el operador AND.

El nuevo uso de este operador es para realizar operaciones matemáticas, como ser: `7+493`

2.6 Bonus

2.6.1 Modo incognito

Siempre que hagas una búsqueda utiliza el navegador en modo incógnito así evitas que los resultados que obtienes estén contaminados por tu historial de navegación y búsquedas.

Capítulo 3

Yandex

Yet **A**nother i**NDEX** (*Índice de idiomas* en ruso) es un Motor de búsqueda para internet que nacio el 23 de septiembre de 1997 cuyos líderes del desarrollo fueron A.Volozh y I.Segalovich. La busqueda se realiza basado en análisis morfológicos (independientemente de los casos y las conjugaciones de palabras en la consulta.[8]

En la figura 3.1 vemos el filtrado de archivos PDF.

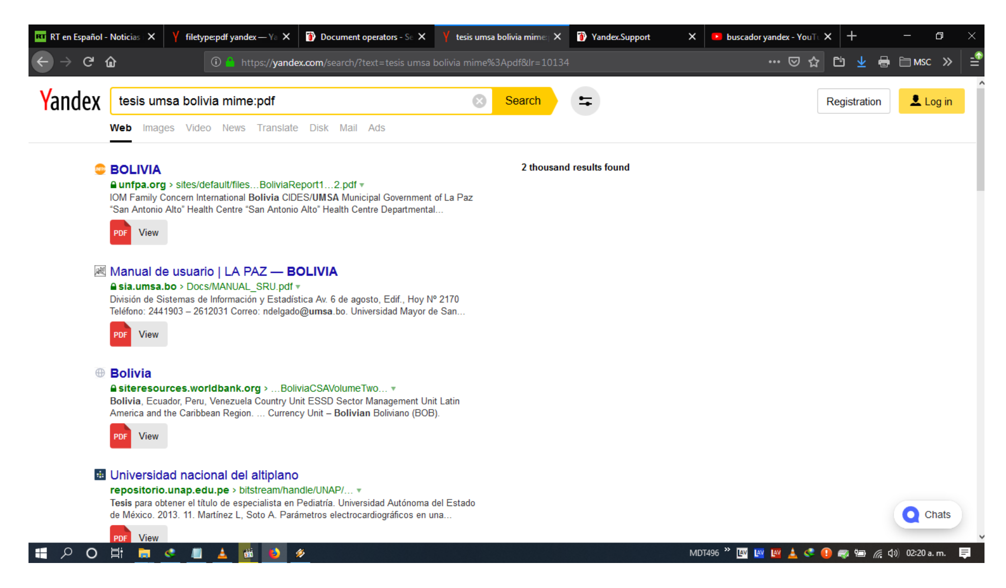


Figura 3.1. Filtrado de archivos PDF

3.1 Operadores

La tabla que sigue (página siguiente) contiene operadores permitidos en yandex, que se obtuvieron del sitio oficial de Yandex Support, en fecha Marzo 2020.

Operadores Yandex

<table>
<tr><th>Operator</th><th>What it does</th><th>Keyword example</th><th>When we show ads</th></tr>
<tr><td><code>!</code></td><td>Matches the form of the word (case, number, and tense).</td><td><code>buy !apples wholesale</code></td><td>✓ buy apples wholesale
✗ buy Apple wholesale</td></tr>
<tr><td><code>+</code></td><td>Matches stop words (auxiliary parts of speech, pronouns, and any words that do not add meaning on their own).</td><td><code>work +from home</code></td><td>✓ work from home
✗ work at home
✗ home work</td></tr>
<tr><td><code>" "</code></td><td>Matches the exact number of words. Ads are triggered by search queries that contain the keyword without any additional words.</td><td><code>"buy cars"</code></td><td>✓ buy cars
✓ car buy
✗ buy red car</td></tr>
<tr><td></td><td></td><td><code>"buy car in showroom in moscow"</code></td><td>✓ buy car in showroom in moscow
✓ buy car in showroom moscow</td></tr>
<tr><td><code>[]</code></td><td>Matches the order of words. This operate will match any of grammatical forms and stop words found inside the [].</td><td><code>tickets [from london to paris]</code></td><td>✓ tickets from london to paris
✓ airline tickets from london to paris
✗ tickets from paris to london
✗ tickets london paris
✗ tickets from london cheap to paris</td></tr>
<tr><td><code>()</code> and <code>|</code></td><td>Groups words in complex search queries.</td><td><code>buy car (cheap|online)</code></td><td>✓ buy car cheap
✓ buy car online</td></tr>
</table>

3.2 Comandos

La tabla que sigue (página siguiente) contiene comandos permitidos en yandex, que se obtuvieron del sitio oficial de Yandex Support, en fecha Marzo 2020.

Comandos Yandex

Operator	Description	Syntax	Example of a query

`url:`	Search pages located at the specified address (URL).
	
	Note. The address is not case sensitive.
	
	To find all documents with addresses that start with the specified value, put the `*` character at the end of the URL.
	
	The URL in the query should be enclosed in quotation marks if it contains one of the following characters:
	`'`
	`"`
	`(`
	`)`
	`_`
		`url:full URL`
		
		`url:host name/category/*`
			[Alice url:en.wikiquote.org/wiki/*]
			
			Documents that contain the name "Alice" and have addresses starting with `en.wikiquote.org/wiki` will be found.

Operator	Description	Syntax	Example of a query
`site:`	Search by all subdomains and pages of the specified site.	`site:website URL`	[site:https://en.wikiquote.org] All documents posted on the subdomains and pages of the website https://en.wikiquote.org will be found.
`host:`	Search pages hosted on this host. Identical to the url: operator with the specified host name.	`host:the host name in the format www.second-level domain.top-level domain`	[search host:www.yandex.com] Documents containing the word "search" and hosted on the `www.yandex.com` host will be found.
`rhost:`	Search pages hosted on this host. Identical to the host: operator, but the host name is written in reverse order: the top-level domain first, then the second-level domain, and so on. To search by all subdomains of the specified domain, put the `*` character at the end of the URL.	`rhost:host name in the format top-level domain.second-level domain.www` `rhost:host name in the format top-level domain.second-level domain*`	[news rhost:com.livejournal.www] Documents containing the word "news" and hosted on the `livejournal.com` domain will be found. [news rhost:com.livejournal.*] Documents containing the word "news" and hosted on all subdomains of the `livejournal.com` domain will be found.
`domain:`	Search pages located on the specified domain.	`domain:domain`	[yandex domain:com] Documents containing the word "yandex" and located on the `com` domain will be found.

Operator	Description	Syntax	Example of a query
`mime:`	Search for documents with the specified file type.	`mime:file type`	[passport application form mime:doc] Documents in the `doc` format containing the "application form" and "passport" words will be found.
`lang:`	Search by pages in the specified language: • Russian (ru) • Ukrainian (uk) • Belarusian (be) • English (en) • French (fr) • German (de) • Kazakh (kk) • Tatar (tt) • Turkish (tr)	`lang: language`	[passport lang:en] The search will return documents in English containing the word "passport".

Operator	Description	Syntax	Example of a query

`date:`	Search by page restricted by the date they were last modified.
	
	Note.
	The year of change must be specified. The month and day can be replaced with the `*` character.
		Page change:
		• corresponds to the specified date:
		`date:YYYYMMDD`
		• is in the range:
		`date:YYYYMMDD..YYYYMMDD`
		• is before or after the specified date (<, <=, >, >=):
		`date:<YYYYMMDD`
		The date of page change is specified in part:
		`date:YYYY**`
			The search will return the documents with the change date:
			• corresponding to 27.07.2018:
			[tiyatro date:20180727]
			• from 27.04.2018 through 27.07.2018:
			[tiyatro date:20180427..date:20180727]
			• after 27.04.2018:
			[tiyatro date:>20180427]
			• corresponding to the year 2018:
			[tiyatro date:2018**]

Capítulo 4

DuckDuckGo

[...] debido a la prevalencia de los rastreadores en los sitios web, los usuarios de la web pueden estar sujetos a perfiles mientras acceden a la web, lo que perjudica su privacidad en línea.[4]

4.1 Introducción

DuckDuckGo, un entrante tardío en la industria de los motores de búsqueda, fue fundado en 2008 por Gabriel Weinberg con el objetivo principal de proteger la privacidad en línea de los usuarios y elevar el estándar de confianza en línea. DuckDuckGo se ganó la reputación de ser "el motor de búsqueda que no te rastrea" y, a pesar de los obstáculos iniciales, en 2017, se había convertido en uno de los 10 principales motores de búsqueda del mundo. DuckDuckGo había logrado aprovechar con éxito las crecientes preocupaciones de privacidad en medio de las acusaciones de seguimiento realizadas contra otros motores de búsqueda[...].[6] Hoy en día, la mayoría del sistema basado en Linux está utilizando DuckDuckGo como motor de búsqueda predeterminado.[7]

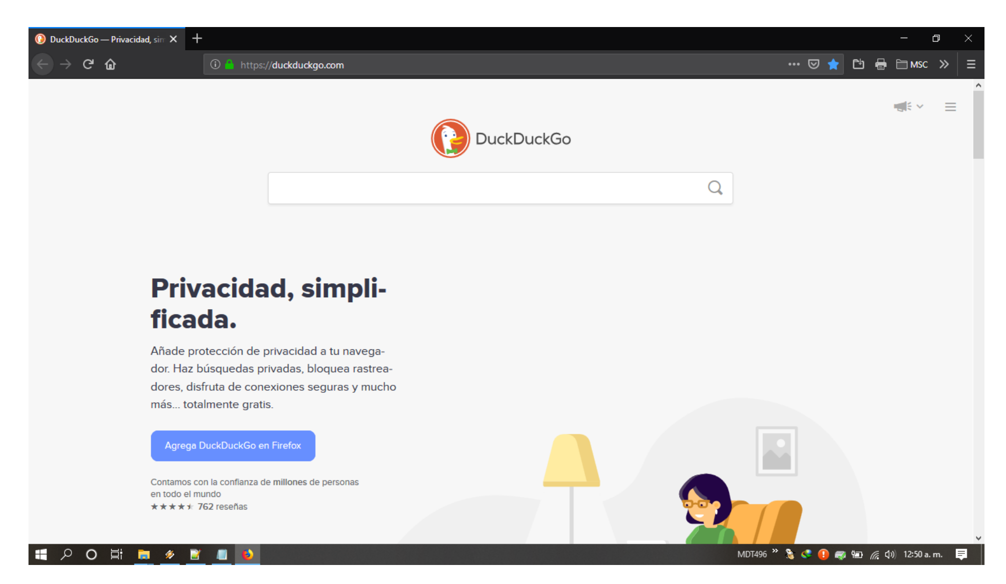


Figura 4.1. duckduckgo

Capítulo 5

Sci Hub

Sci-Hub es una creación de Alexandra Elbakyan, una estudiante graduada de Kazajstán, quien lo fundó en 2011 como respuesta a las restricciones financieras que impidieron que ella y sus colegas investigadores obtuvieran libremente literatura academica.[2]

Sci-hub es un [...] sitio web con millones de artículos de investigación para estudiantes y académicos. Todo el papel que proporciona es casi libre [...] desafortunadamente, es una manera ilegal de descargar documentos, Sci-hub sigue cambiando su nombre de dominio. [3]

En la figura 5.1 vemos una demostración de dicha herramienta, que dado una url (Uniform Resource Locator) o su DOI (Digital Object Identifier)[1], retorna el artículo en cuestion.

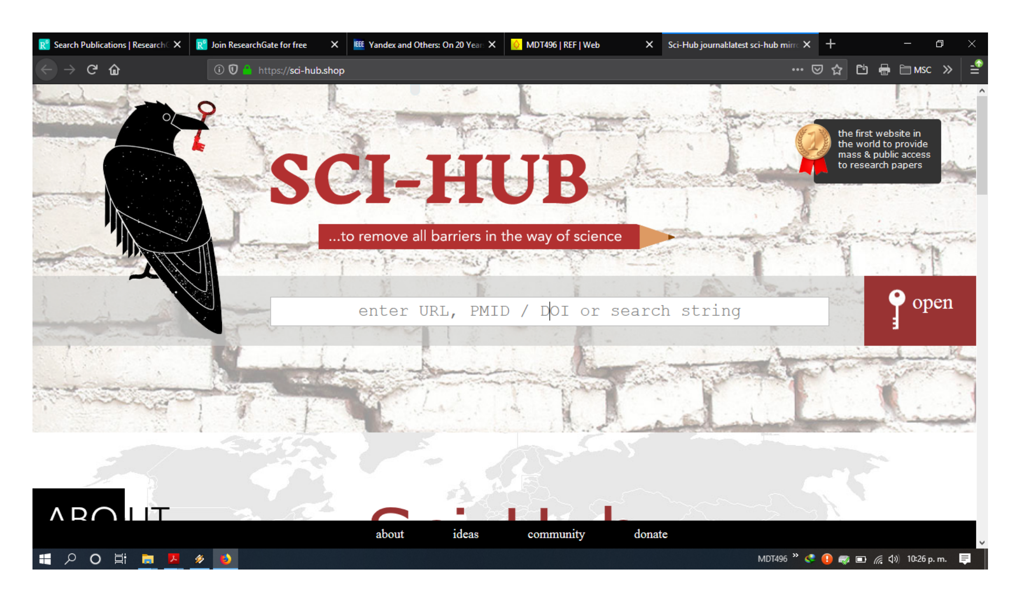


Figura 5.1. Sci-hub

[1]Un DOI es un identificador único y permanente para publicaciones electrónicas, por ejemplo un artículo electrónico de una revista.

Referencias

[1] José María Alonso Cebrián, Antonio Guzmán Sacristán, Pedro Laguna Durán, and Alejandro Martín Bailón. *Ataques a aplicaciones web*. PID 00191662. FUOC. Fundación para la Universitat Oberta de Catalunya, 2012.

[2] William Badke. "Sci-Hub and the Researcher". In: *Online Searcher* 41 (Jan. 2017), pages 56–58.

[3] Ashish Chaurasiya. *Best Sci-hub Alternatives to Download research Papers for Free or Sci-Hub (Latest alternative links)*. Apr. 2020. DOI: 10.13140/RG.2.2.25156.78723.

[4] Matthew Corner, Huseyin Dogan, Alexios Mylonas, and Francis Djabri. "A Usability Evaluation of Privacy Add-ons for Web Browsers". In: July 2019, pages 442–458. ISBN: 978-3-030-23534-5. DOI: 10.1007/978-3-030-23535-2_33.

[5] Institute for Security and Open Methodologies. *Hacker Highschool*. 2004. http://www.hackerhighschool.org/license.

[6] Nakul Parameswar, Jahnavi Maddipatla, and Sanjay Dhir. "Ivey Publishing Case Study - DUCKDUCKGO: REINVENTING THE INTERNET SEARCH". In: (June 2019).

[7] Dr Parsania. ""A Comparative Analysis: DuckDuckGo Vs. Google Search Engine"". In: *Global Research and Development Journal for Engineering | Volume 2 | Issue 1 | December 2016 ISSN: 2455-5703* 2 (Dec. 2016), pages 12–17.

[8] Yuri Polak. "Yandex and Others: On 20 Years of Russian Search in Internet". In: Oct. 2017, pages 172–175. DOI: 10.1109/SoRuCom.2017.00034.

[9] Wikipedia. *Secure Hash Algorithm — Wikipedia, La enciclopedia libre*. 2020. https://es.wikipedia.org/w/index.php?title=Secure_Hash_Algorithm&oldid=123676933.

Glosario

M

Message Digest 5 (MD5) Algoritmo desarrollado en 1991 utilizado para crear firmas digitales. MD5 es una función de una vía, toma un mensaje de longitud variable y lo convierte en una salida de longitud fija conocido como digestión. Su intención es para utilizarse con máquinas de 32 bits. Es mas seguro que el algoritmo MD4, el cual ya ha sido comprometido.[5], pág. 1.

S

Secure Hash Algorithm (SHA) El SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es una familia de funciones hash publicadas por el Instituto Nacional de Normas y Tecnología, INNT (NIST en idioma inglés) de Estados Unidos. La primera versión del algoritmo se creó en 1993 con el nombre de SHA, aunque en la actualidad se la conoce como SHA-0 para evitar confusiones con las versiones posteriores.[9], pág. 1.